Computer Design Strategy for MCM-D/Flip-Chip Technology

Paul D. Franzon Tom Conte Sanjeev Banerjia Scott Alvorez
Alan Glaser Steve Lipa Toby Schaffer Andrew Stanaski
Yusuf Tekmen

Invited Paper – Extended Abstract

Department of Electrical and Computer Engineering, Box 7911
North Carolina State University, Raleigh, NC 27695

Abstract.

A compelling case is made for using MCM-D (thin film MultiChip Module) flip-chip technology to build a 'MegaChip' CPU consisting of an Instruction Fetch Unit and Execution Unit. By building part of the Instruction Fetch Unit in an optimized SRAM process, significant performance/cost gains are made. We also address the following important 'implementation' (1) Partitioning high speed paths across the chip boundary within timing specs; (2) Ability to use off-the-shelf memories; (3) Using the MCM for power, ground, and clock distribution; (4) Managing test costs; and (5) implementing a debug strategy. This paradigm is also potentially useful for other memory intensive applications, including ATM, etc.

1 Introduction

In this paper, we investigate the application of a dense MultiChip Module (MCM) technology (see Figure 1) to improving computer performance and reducing implementation cost. (Previous work is published in references [1, 2].) The main advantage to be gained through using MCM technology is that of optimizing the technology mix. SRAM built in an SRAM process is about 50% denser and 50% faster than SRAM built in a modified logic process. Thus there is tremendous advantage to building the caches in a computer in an SRAM process and using an MCM to integrate the cache and Execution Unit (EU). However, this advantage is largely lost if the timing across the cache-EU interface becomes unwieldy or if test and debug difficulties make the design execution too difficult or expensive.

In the rest of this paper, we explore a number of issues related to this paradigm: (1) A 'producer-consumer' architecture that maximizes the benefit of the technology mix, (2) a detailed case study of the inter-chip timing issues, (3) the benefits of using the thin film MCM layers for global power/ground and clock distribution, and (4) managing test and debug issues.

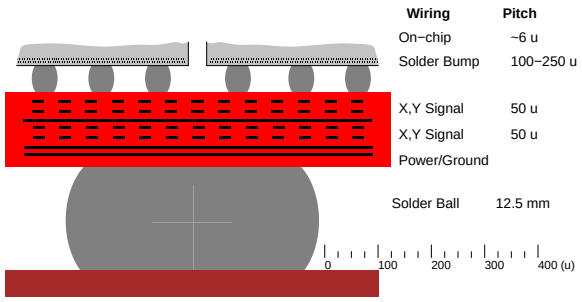

Figure 1: High Density MCM-D/flip-chip technology being targeted.

2 An Instruction Fetch Unit/Execution Unit Microarchitecture

In the proposed organization, incoming code from the instruction cache is dynamically scheduled into VLIW-like multi-ops (MOPS) by the schedule unit and is placed in the schedule cache, which then serves as the first-level cache for the execution unit. Since the schedule unit and schedule cache are designed to match the "resource width" of the execution unit, the schedule cache can dispatch one line of scheduled code per cycle. This becomes interesting at large issue widths, when the schedule unit is required to speculate across multiple branches to match its issue width to that of the execution unit. This approach lies somewhere between superscalar and VLIW architectures, and can be tuned to mimic either.

The compelling factor in this proposal is that multi-die technology is leveraged to produce a high-

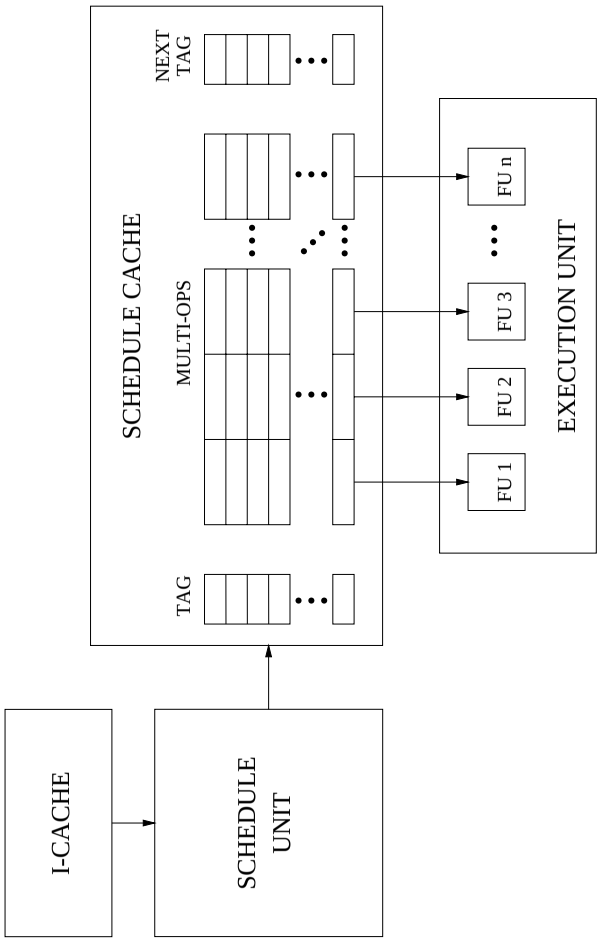

Figure 2: An Instruction Fetch Unit/Execution Unit Microarchitecture

performance solution. The various units are fabricated in whatever process is appropriate (e.g. the execution unit is fabricated in a logic process, while the schedule cache is fabricated in an SRAM process), which optimizes both area and speed for each chip. The result is that all of the units may be driven at a high clock rate (the execution units in particular), while more resources may be devoted to each unit. As an example, the number of available functional units may be increased over extant microprocessors by the simple fact that cache memory is removed from the die; the higher density of memory built in an SRAM process implies that the schedule cache may be made very deep or may be multi-banked.

3 Maintaining Timing Across Chip Boundaries

A possible MCM floorplan is shown in Figure 2.

The most aggressive aspect of the this partition is that the I–Cache access must be performed in a single

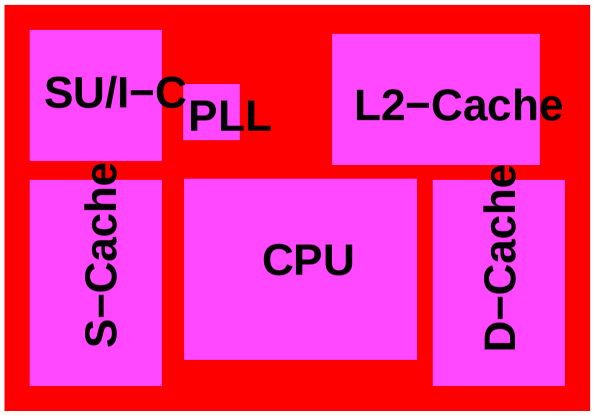

Figure 3: A feasible partitioning for the optimized processor core.

clock cycle. To achieve a single cycle fetch, it is necessary to floorplan the I–Cache and CPU and determine solder bump locations so that the interconnect delay from the instruction unit to the address logic, and from the sense amplifier back to the instruction unit, is only a small part of the clock cycle. The potential danger is that breakout from the on-chip cell through the solder bump to the routing channel might increase the path length, and thus the delay, by too much.

Figure 3 shows a floorplan that will minimize the effect of the breakout on clock cycle time. Figure 4 shows part of the breakout pattern from either the SRAM or the CPU chip. In this breakout pattern, the solder bumps are on a 200 μm pitch. For each end of the pattern, there are 228 signal I/Os and 228 power/ground pins. With 4 signal layers, eight rows of signal pins can break out in one direction (only four rows and the top layer of routing is shown in Figure 4). Thus an array of 29 by 16 solder bumps is needed, taking up 5.6 $\times$ 3.2 mm.

As this solder bump array is larger than the I/D unit, on-chip routing will be required for the breakout on the CPU side. Early estimates indicate that up to 3 mm might be needed.

In the SRAM, a 32 by 14 solder bump array is used, consuming 6.4 $\times$ 2.8 mm of area. The longest on-SRAM connection (to a sense amplifier) is about 2 mm.

The longest on-MCM path in this connection is 9 mm. The I–Cache to Instruction/Decode unit input register path thus has a total length of 14 mm. This path is driven by a single driver, and simulation results indicate that the total delay would be about 600 ps. In comparison, the I–Cache to Instruction/Decode unit input register delay was about 100 ps before partitioning. Crosstalk, at 3.2%, is not a problem in this signal path.

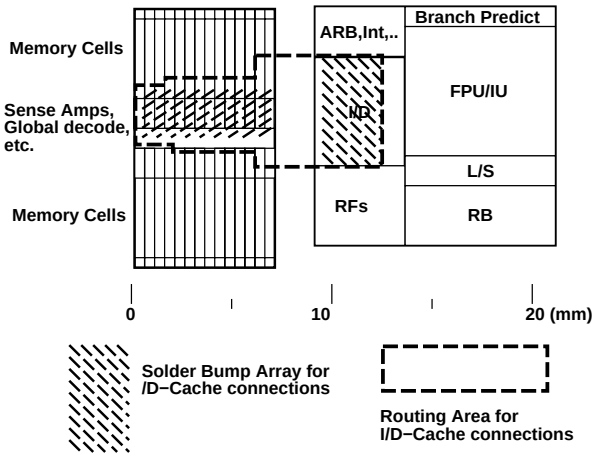

Figure 4: CPU and I–Cache chip floorplans. The Instruction/Decode unit communicates with the I–cache.

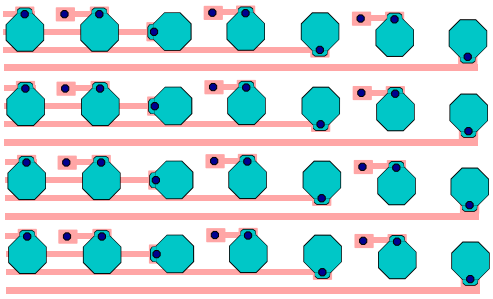

Figure 5: Details of part of the breakout pattern for the I/D to I–cache connection. Only the top signal layer is shown. Another, lower signal layer permits a similar array of bumps to the right of these to be broken out.

Not shown in these figures are the I–Cache to Level-2-Cache refill connections. Though not a critical path, these connections need to be 196 bits wide and thus will consume considerable interconnect resources both on the I–Cache and on the MCM. These would be placed below the pads shown on the I–Cache.

In conclusion, this partitioning is feasible, though with a 250 ps latency penalty. This penalty is more than compensated by the speed-up obtainable by building the caches in an SRAM technology. In further work, we plan to build part of these circuits and more precisely determine the optimal floorplan and circuit structures.

4 Using the MCM for Global Clock, Power and Ground Distribution

Interconnect traces on an MCM-D substrate behave like lossy transmission lines as opposed to the RC behavior of on-chip interconnect. This difference in behavior leads us to explore the distribution of global clock signals via MCM interconnect rather than IC interconnect. In addition, using an MCM substrate with area-array solder bumps provides multiple entry points distributed over the IC circuitry, many more than would be available using standard peripheral pads. This presents the opportunity not only to dramatically increase the I/O capacity but also to provide local power and ground through the bumps rather than global power and ground via on-chip rails. (Small on-chip global power and ground connections might still be required to provide a path for signal return currents.) The term "local" is loosely defined – it depends upon the fraction of bumps available for power and ground connections; this is discussed further below.

As a vehicle for these experiments, we have designed a full-custom implementation of the ANSI Data Encryption Algorithm, commonly referred to as DES, in a 0.6μm, three-metal silicon technology. The chip consists of 16 identical *rounds* as shown in Figure 6. Data flows in in a "U" shape, where each round is a pipeline stage. To minimize the interconnect length between rounds eight and nine (the bottom of the "U"), the two columns must be skewed, i.e., rounds are not aligned horizontally.

4.1 On-MCM Clock Distribution

Standard on-chip clock distribution nets, e.g., H–trees, suffer from the RC parasitics of on-chip wiring in addition to consuming large amounts of routing capacity on the metal layer used. Routing the clock net on the MCM substrate, and providing local entry points on-chip through the solder bumps at the leaves of the clock tree, effectively eliminates one stage (driver and wiring) in the on-chip clock distribution. For our design example, we simulated the entire clock distribution (including chip attachment parasitics) using an on-MCM H–tree with 16 leaf entry points (one for each round) and local on-chip distribution after each entry point. We designed the driver for the clock distribution network on the MCM substrate by appropriately sizing it, so that a good clock edge is available at the entry points (the driver giving the 'squarest' waveform in Figure 5). It was *not* necessary to size the line widths in the H–tree; we used minimal line width for all branches, greatly reducing the effect of the H–tree on the routing underneath the chip. The H–tree itself is insensitive to

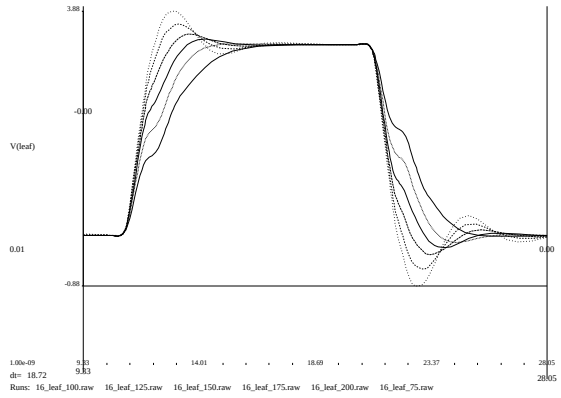

Figure 6: Clock waveforms at chip entry points for different driver sizes.

load mismatches at the leaves; an extreme case where one of the leaves was loaded twice as much as the others resulted only in 10 ps skew. This implies that a virtually skewless clock can be guaranteed at the entry points, if the distribution network is carefully designed. On-chip drivers are located in the vicinity of the entry points and deliver the clock signal to balanced loads. Skew is further reduced by removing one layer of on-chip clock buffering.

4.2 On-MCM Power and Ground Distribution

In a conventional peripheral pad design, large power and ground rails must reach every part of the chip. As shown in Figure 6, there is a power rail on either side of the logic and a ground rail in the middle. Horizontal metal fingers then provide power and ground to the individual rounds.

The situation is different in the case of local power distribution through area-array solder bumps. As noted above, the definition of "local" power and ground distribution is design-dependent; in our case, the density of the solder bumps is sufficient to allocate one bump each for power and ground per round. (This capacity is available even after all I/O and clock signals have been accounted for.) While distributing power and ground in this manner does not change the size of the horizontal fingers, it eliminates the need for the vertical rails altogether, as shown in Figure 7. For this example, the area saved by removing these rails is approximately 4.67 sq. mm, or a 34% reduction. In both cases, the maximum peak IR drop meets the design requirement of being 5% or less.

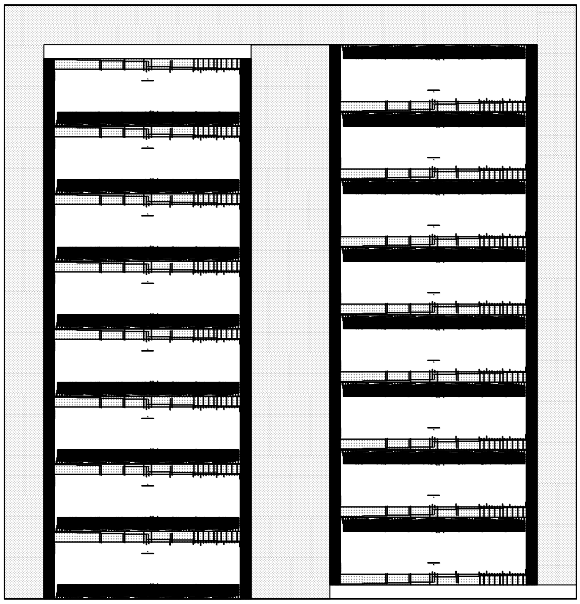

Figure 7: Floorplan using conventional (peripheral-pad) power distribution.

5 Managing Test Costs and Verification

In light of the the preceding discussion, cost effective methods for testing bare, high I/O chips that have been electrically designed specifically for use on an MCM are needed. These methods must be able to prove the chips function correctly and determine the speed rating of each chip. In addition, it is desirable to use existing VLSI testers to perform the tests.

The physical interface issue is being addressed through the development of new test head technologies, such as membrane probes. Ideally the probes and chips would be designed so one probe could be used to test all the dice. However, in a high part count situation, the additional tooling cost of unique probe heads would only add a small overhead. A more significant problem is matching the chip electrical environment to the tester electrical environment. The chips, designed to run exclusively on the MCM, will not have enough drive strength to communicate a fast signal to the tester. Also the number of I/Os from the chip is likely to be larger than the tester can accommodate. Thus some active circuitry, drivers and multiplexers at least, will be needed on the test head.

Placing active circuitry on the test head presents an opportunity to extend the capability of the VLSI tester at low cost. In addition to drivers and multiplexers, pattern generation and compaction as well as IDDQ measurement can be incorporated in the test

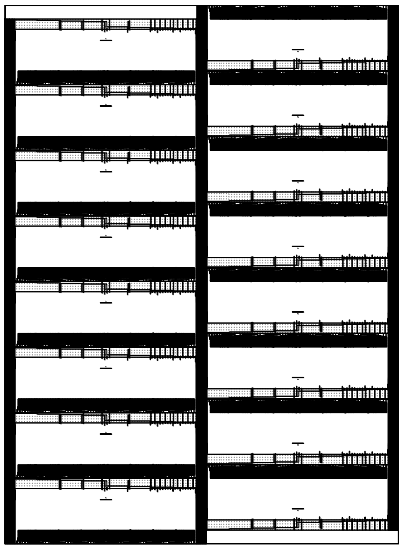

Figure 8: Floorplan using solder-bump power distribution.

head. This allows the test head to perform some of the testing autonomously and reduces the memory requirement of the VLSI tester. Also, since the test head can be running independent of the tester for a period of time, multiple test heads can share the same tester. Thus, a chip designed for MCM implementation will require less tester memory and lower test time, at the expense of using a smart test head.

There is another issue for flip-chip MCM systems: failure diagnosis, particularly of delay faults. When a chip is flipped it is no longer accessible for voltage contrast probing. Failure diagnosis capability can be improved by the use of full scan. Sun uses this effectively on their SuperSparc 2 and UltraSparc chips [4, 3]. Also, voltage contrast probing can be accommodated in the test head by etching holes in the membrane probes. By making a set of membrane probes together with chip versions with different pad locations, it should be possible to provide 100% access. This provides the needed coverage during the development process. When the chip is mature, a single probe and chip arrangement is adequate for functional and speed testing.

Acknowledgments

The authors wish to thank the following funding and support sources: ARPA under contract DASH04-94-G-003-P2, NSF under grants MIP-901704 and DDM-9215755, NSF for Dr. Franzon's NSF Young Investigator's Award and Toby Schaffer's NSF Graduate Fellowship, and Intel Corporation. The authors also wish to thank Robert Frye, Thad Gabara, Wayne Nunn, Steve Mozgai, Bill Schmidt, Frank Swiatoweic, Evan Davidson, and Gary Dudeck.

References

[1] Sanjeev Banerjia, Alan Glaser, Christoforos Harvatis, Steve Lipa, Real Pomerleau, Toby Schaffer, Andrew Stanaski, Yusuf Tekmen, Grif Bilbro, and Paul Franzon. Issues in partitioning integrated circuits for mcm-d/flip-chip technology. In *1996 IEEE MultiChip Module Conference*, pages 154–160, 1996.

[2] P. Dehkordi, K. Ramaurthi, D. Bouldin, and H. Davidson. Early cost-performance cache analysis of a split mcm-based microsparc cpu. In *1996 IEEE MultiChip Module Conference*, pages 148–153, 1996.

[3] H. Hao and R. Avra. Structured design-for-debug - the supersparc ii methodology and implementation. In *Proceeings of the 1995 International Test Conference*, pages 175–183, 1995.

[4] M. Levitt, S. Nori, S. Narayanan, G. Grewal, L. Youngs, A. Jones, G. Billus, and S. Paramanandam. Testability, debuggability, and manufacturability features of the ultrasparc-i microprocessor. In *Proceeings of the 1995 International Test Conference*, pages 157–166, 1995.